Natural Language Processing

Artificial Intelligence

Mariia Korol

Data Science Major
Montana Tech

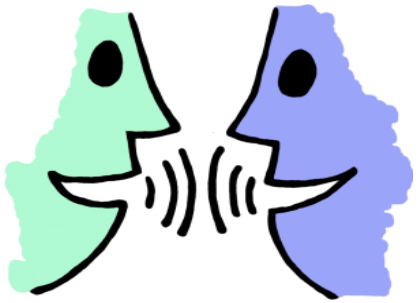
Outline

What is NLP?

Computer Science

Data Science

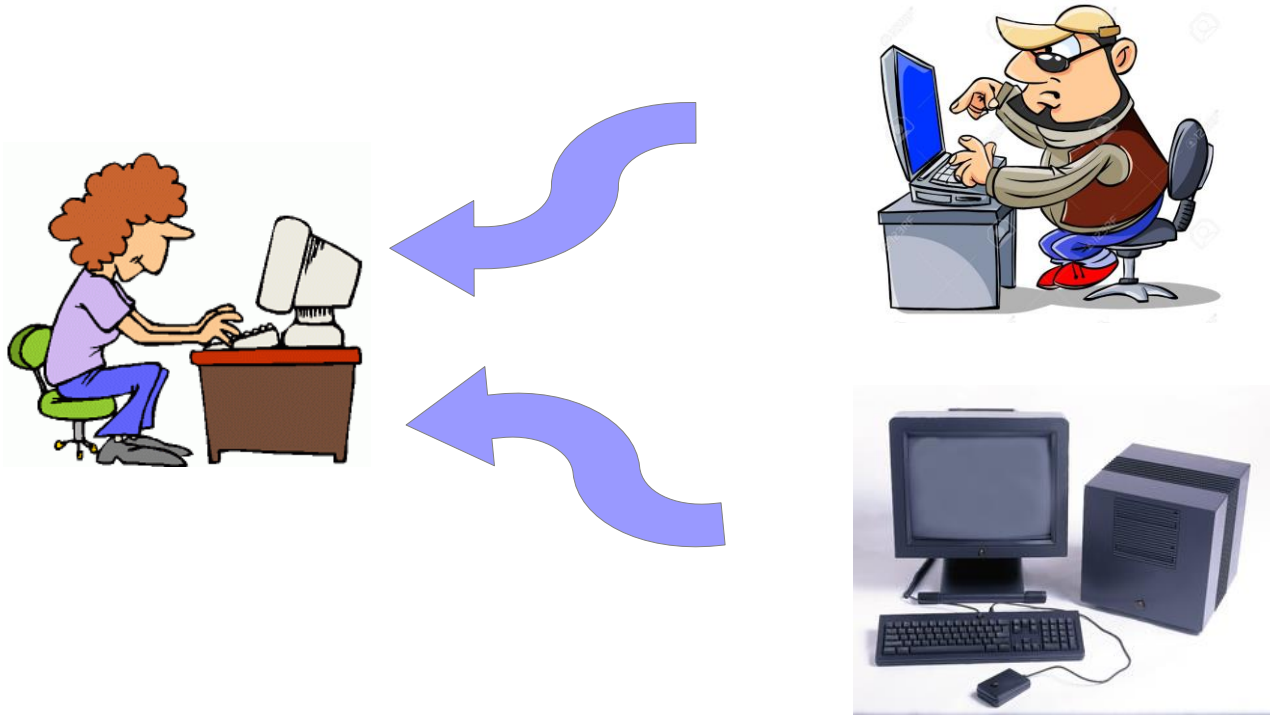
NLP (Natural Language Processing)
computers dealing with language



Linguistics

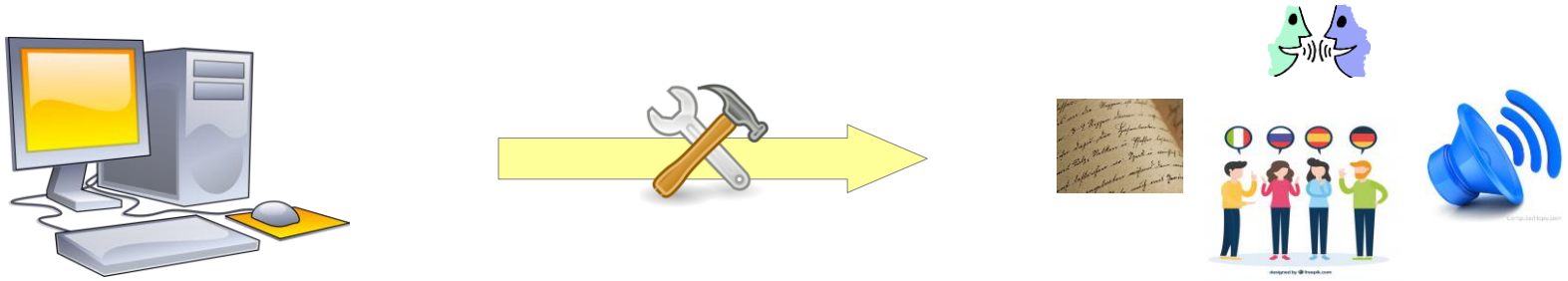
Turing Test

History flash back of NLP: Test of Alan Turing in 1950s



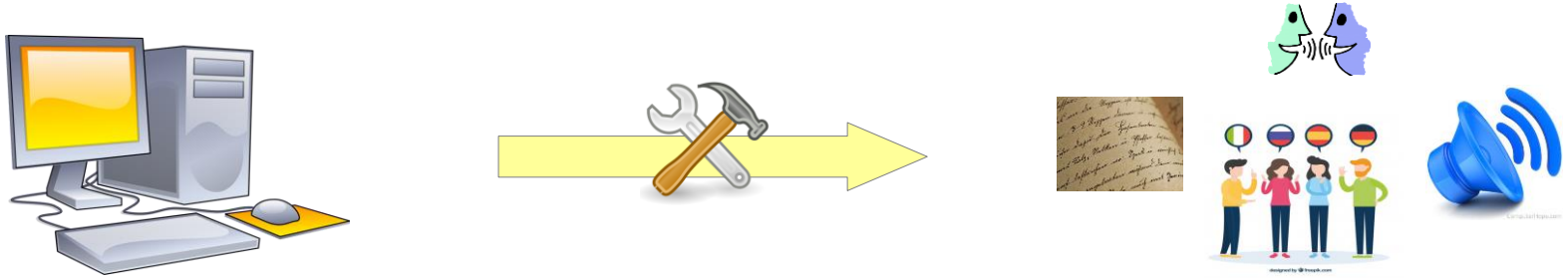
Can a human distinguish between texting with another human and a computer program?

Applications of NLP



- Language translation applications such as **Google Translate**
- Word Processors such as **Microsoft Word** and **Grammarly** that employ NLP to check grammatical accuracy of texts.
- Interactive Voice Response (**IVR**) applications used in call centers to respond to certain users' requests.
- Personal assistant applications such as **OK Google, Siri, Cortana**, and **Alexa**.

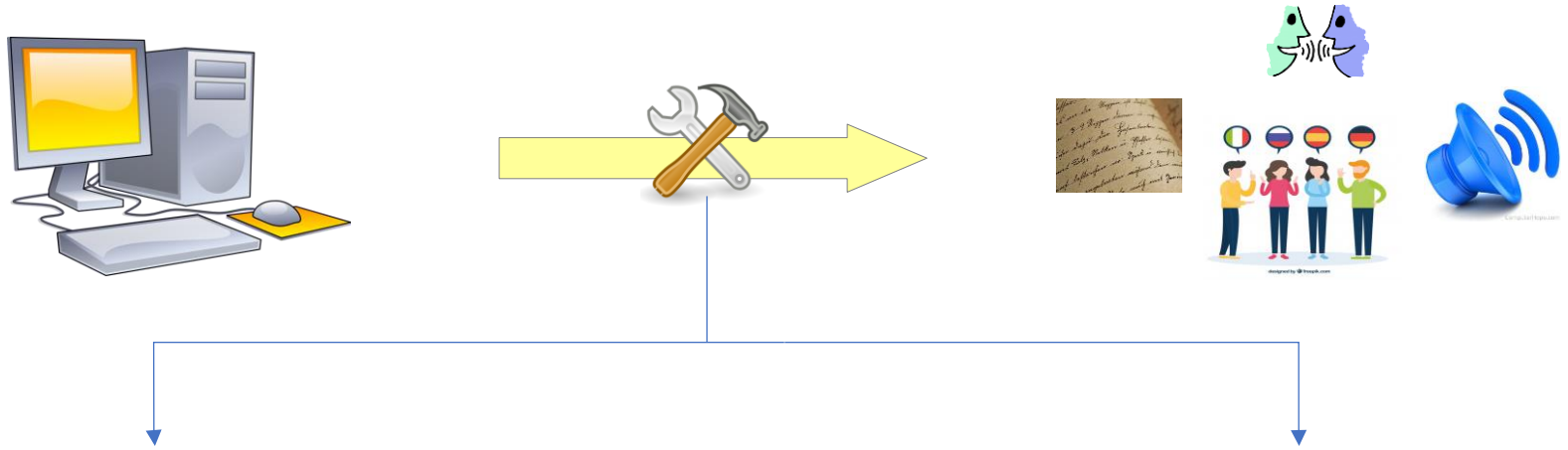
Applications of NLP



- **Statistical text/document analysis**: classification, clustering, search for similarities, language detection, etc
- **Capturing syntactic information**: part-of-speech tagging, chunking, parcing, etc.
- **Captuting semantic information (meaning)**: word-sense disambiguation, semantic role labelling, named entity extraction, etc.

The presentation concentrates on **text similarity search** and **text clustering**

Approaches to NLP



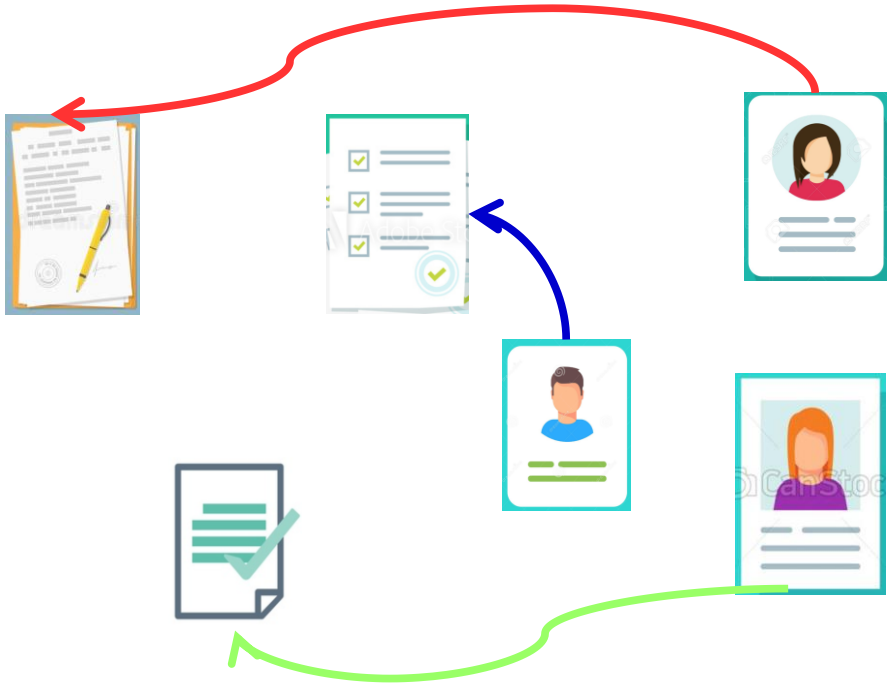
Rule Based Approach

- **Harcoded rules** based on some knowledge
- **Simple**
- **Robust**
- **Not flexible**

Statistical Approach

- **Statistical algorithm** which search patterns and rules
- **Flexible**
- **Generic**
- **Complex**

Text Classification and Similarity



- Finding similar texts by content
- Assigning texts to predefined categories
- Finding clusters of texts by content

What is needed?

- Represent the texts in computer readable format (numbers)
- Run statistical algorithms for these texts

Bag of Words Algorithms

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

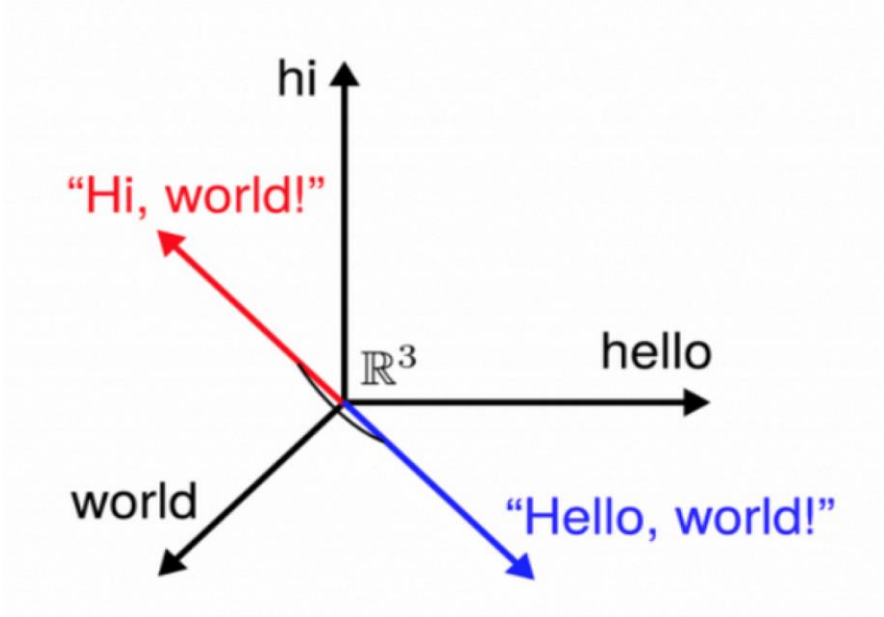
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1

The order and the meaning of the words do not matter

TF – IDF Algorithm

TF: Term Frequency

Transform texts into numeric vectors, where each unique word is a separate dimension



Text 1: Hi, world
Text 2: Hello, world

↓

Coordinate phase space: Hi, Hello, World

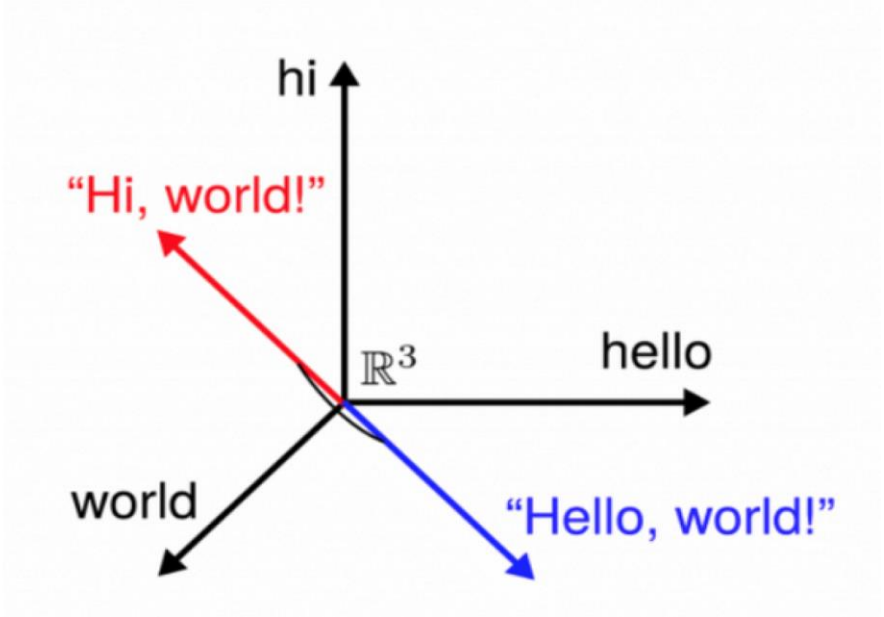
↓

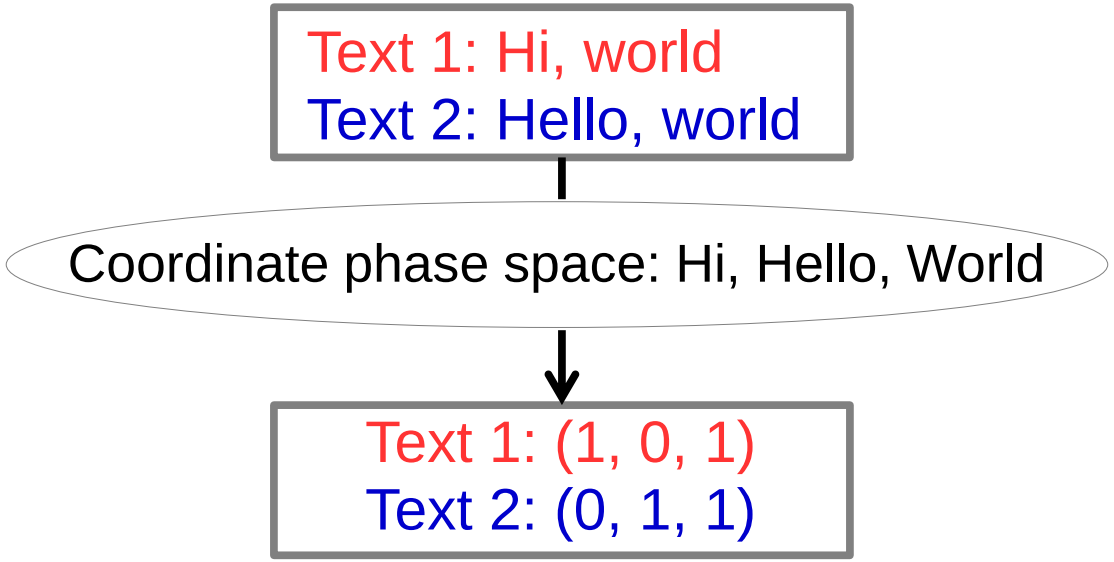
Text 1: (1, 0, 1)
Text 2: (0, 1, 1)

TF – IDF Algorithm

TF: Term Frequency

Transform texts into numeric vectors, where each unique word is a separate dimension



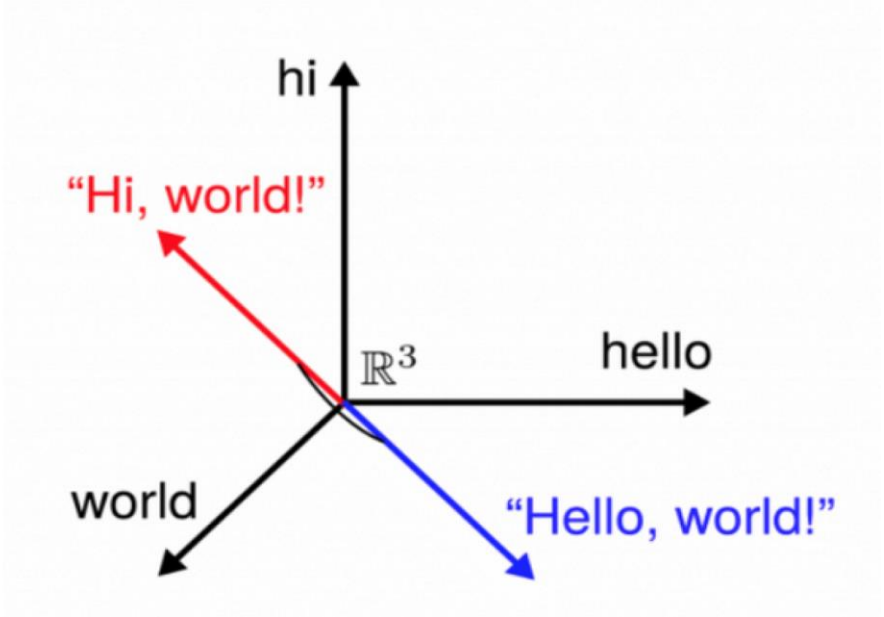


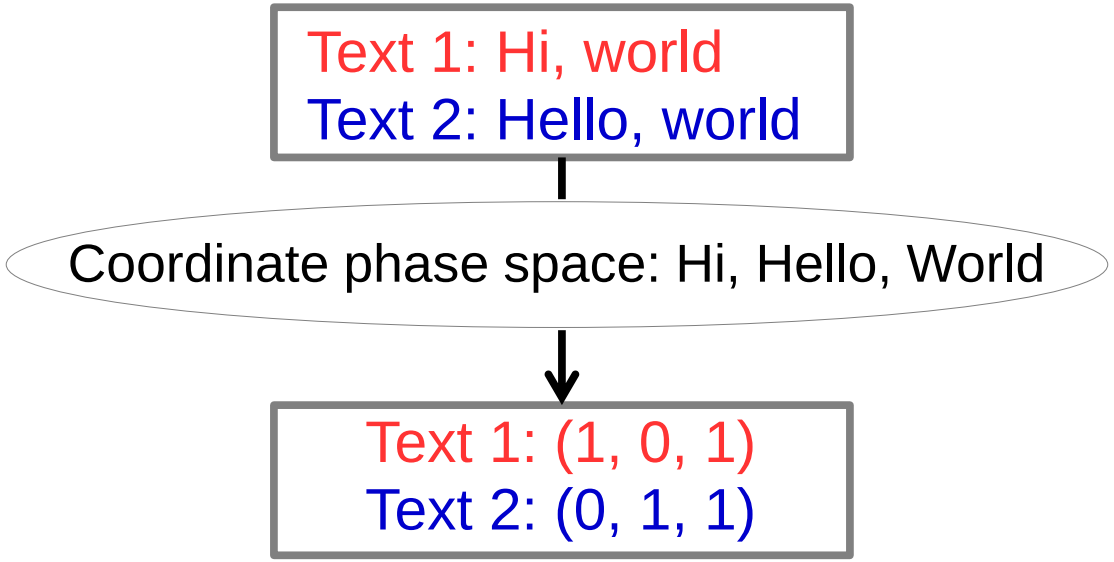
Zipf's law: $f \sim r^{\alpha}$

TF – IDF Algorithm

TF: Term Frequency

Transform texts into numeric vectors, where each unique word is a separate dimension





Zipf's law: $f \sim r^{\alpha}$

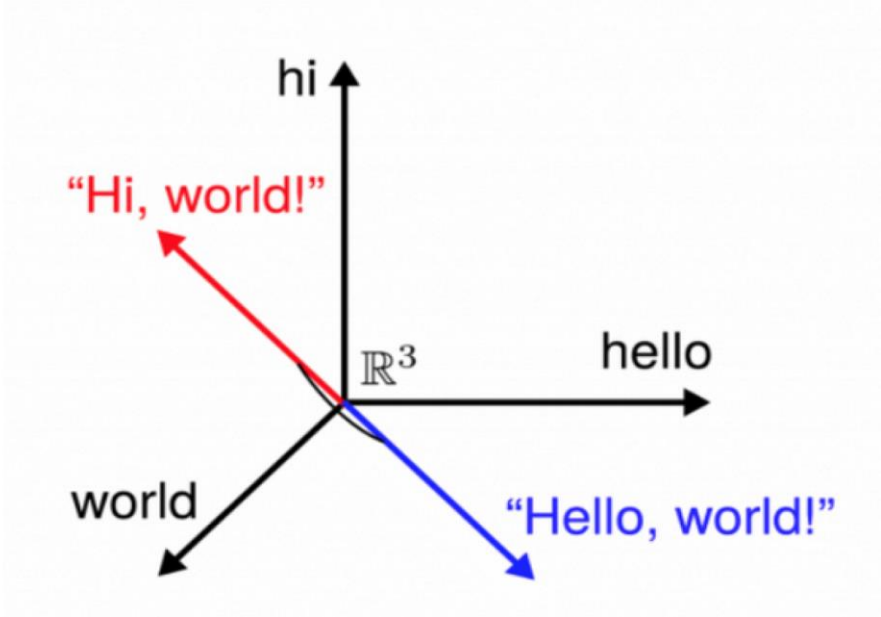
relevancy of a word to a document

$$\frac{\text{Term Frequency}}{\text{Document Frequency}}$$

TF – IDF Algorithm

TF: Term Frequency

Transform texts into numeric vectors, where each unique word is a separate dimension



Text 1: Hi, world
Text 2: Hello, world

↓

Coordinate phase space: Hi, Hello, World

↓

Text 1: (1, 0, 1)
Text 2: (0, 1, 1)

IDF: Inverse Document Frequency

Penalize words which are frequent but don't have any meaning: a, the, is, etc.

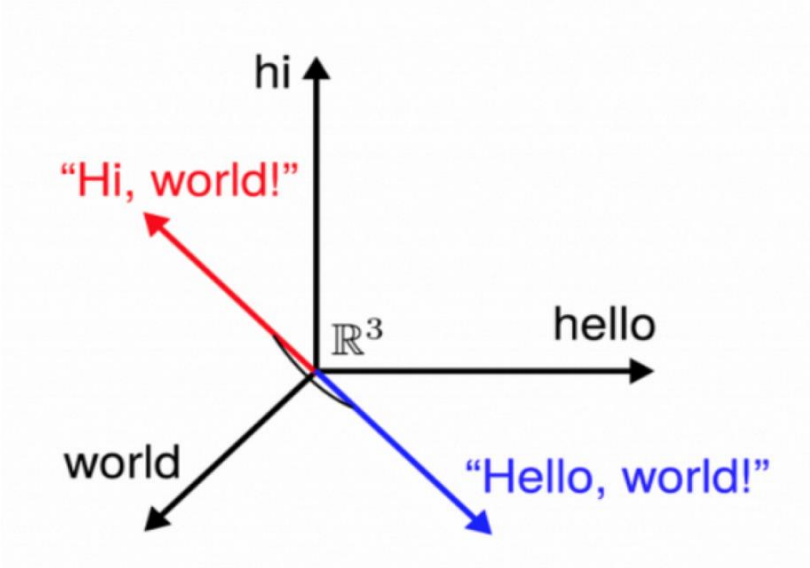
Each TF coordinate is multiplied with a weight

$$\omega_i = \log\left(\frac{N}{DF}\right)$$

- N: Total number of texts
- DF: Number of texts in which a certain word appears

Cosine Similarity

Text 1: Hi, world
Text 2: Hello, world



Are the texts similar?

Look at the angle between the vectors
If vectors are collinear – texts are similar
If vectors are orthogonal – texts are different

Measure – cosine

Orthogonal. But similar!

Hello and Hi different words
but same meaning!

Cosine Similarity

$$\cos(\overrightarrow{TFIDF1}, \overrightarrow{TFIDF2}) = \frac{(\overrightarrow{TFIDF1} \cdot \overrightarrow{TFIDF2})}{\|\overrightarrow{TFIDF1}\| \cdot \|\overrightarrow{TFIDF2}\|}$$

Text 1: (1, 0, 1)
Text 2: (0, 1, 1)

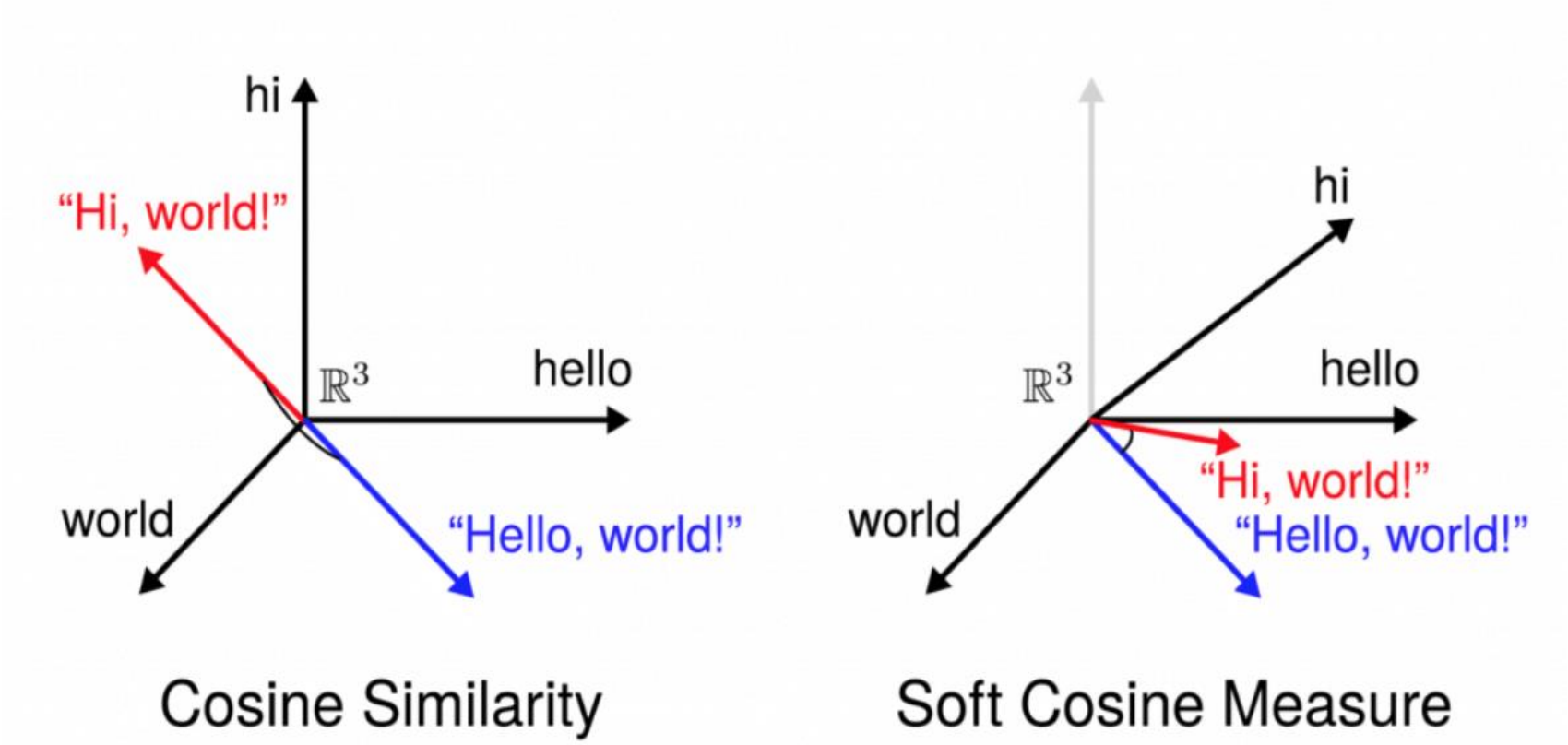
$$(\overrightarrow{TFIDF1} \cdot \overrightarrow{TFIDF2}) = \sum TFIDF1_i \cdot TFIDF2_i$$

$$\| \overrightarrow{TFIDF1(2)} \| = \sqrt{\sum_i TFIDF1(2)_i{}^2}$$

$$\frac{[1, 0, 1] * [0, 1, 1]^T}{\text{sqrt}(1+0+1) * \text{sqrt}(0+1+1)}$$

Soft Cosine Similarity

$$\cos_{soft}(\vec{a}, \vec{b}) = \frac{\sum_{i,j} s_{ij} a_i b_j}{\sqrt{\sum_{i,j} s_{ij} a_i a_j} \cdot \sqrt{\sum_{i,j} s_{ij} b_i b_j}}$$



Clustering of Texts

How to Cluster Texts by Topic?

- Represent texts in **IT-IDF vectors**
- Set a **threshold of similarity** on soft cosine similarity measure
- **Select groups of texts** which are similar within the selected threshold

Or make use of Clustering

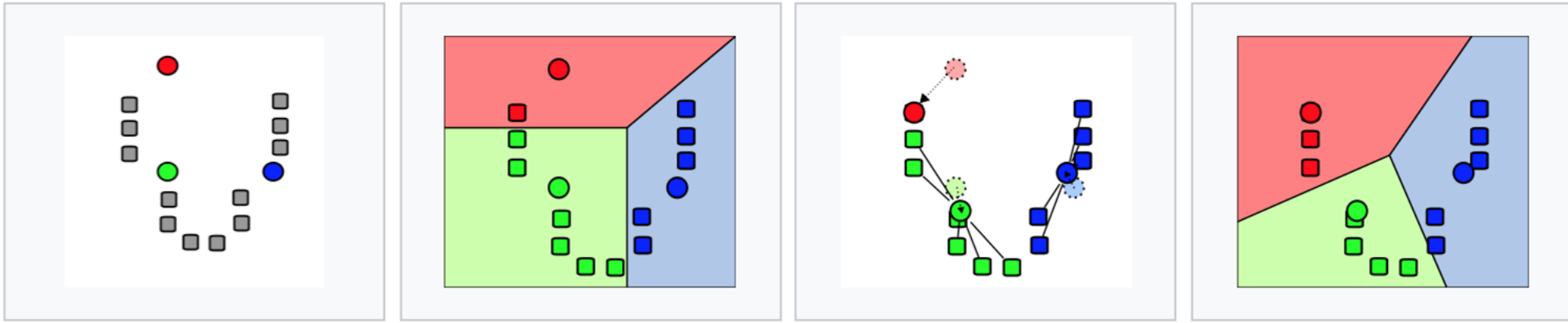
K Means Clustering

Basic K-Means Algorithm

Choose k number of clusters to be Determined.
Choose k objects randomly as the initial cluster center
Repeat
    Assign each object to their closest cluster
    Compute new clusters (calculate mean points)
Until No changes on cluster centers (centroids do not change location any more)
  OR No object changes its cluster

K Means Clustering

Base – TF-IDF vectors



The overall distance to geometrical centers of the clusters is minimized

$$V = \sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2$$

Summary

- NLP – very wide field which combines computer science, linguistics and data science
- NLP finds its applications in every field where language is applied in any form
- This work was concentrated on text similarity search and text clusterind
- IT-IDF algortihm can represent texts in form of numeric vectors
- Soft cosine similarity is an intuitive but efficient method to find similarity between texts
- Any clustering algorithm can be applied upon IT-IDF vectors. One of the most simplest, but widely used algorithms is K Means clustering

References

Deokar, S. T. (2013). Text Documents clustering using K Means Algorithm. International Journal of Technology and Engineering Science, 1(4), 282–286. Retrieved from https://pdfs.semanticscholar.org/4a43/dc3e76082aef3c1fa920b5d023dbf2cb3571.pdf

Garbade, M. J. (2018, October 15). A Simple Introduction to Natural Language Processing. Retrieved from https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32

Yu, S., Xu, C., & Liu, H. (2018). Zipf's law in 50 languages: its structural pattern, linguistic interpretation, and cognitive e motivation. Retrieved from https://arxiv.org/abs/1807.01855

Machinelearningplus.com. (2018, October 30). Cosine Similarity - Understanding the math and how it works? (with python). Retrieved from https://www.machinelearningplus.com/nlp/cosine-similarity/.

Wang, Y.X. (2019, January 29). Artificial Intelligence. Retrieved from https://sites.cs.ucsb.edu/